

UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering
EEC 172 SQ 2021

LAB 2: Serial Interfacing Using SPI
Due: April 18, 2021

Objective: In this lab, you will use SPI to interface the CC3200 LaunchPad to a color OLED display.

New Equipment Needed

- 2 Adafruit OLED Breakout Board - 16-bit Color 1.5" (product id: 1431) on small breadboards
- 1 Saleae USB Logic Analyzer (**optional**)
- Jumper wires

Note: Do not lose the shorting blocks (jumpers) that come with the board. They are used to configure different hardware functions on the board and may be needed for this and future labs. Unused shorting blocks should be kept on the board attached to a single header pin.

Resources (Technical Data Sheets)

Refer to the following technical documents.

- CC3200 Technical Reference Manual
- CC3200 LaunchPad User Guide
- CC3200 LaunchPad Schematic
- CC3200 Datasheet
- SSD1351 128 RGB x 128 Dot Matrix OLED/Common Driver with Controller Advance Information • OLED Display Module Product Specification (Doc No. SAS1-D038-A)

Part I. Interfacing to the OLED using the Serial Peripheral Interface (SPI)

In this part, you will interface to the 128x128 color organic light-emitting diode (OLED) display to the CC3200 LaunchPad using SPI.

1. Implement SPI_demo project on CC3200 LaunchPads

As a starting point, you should study, import, build, and test the modified version of spi_demo project on one CC3200 LaunchPads. Please, replace the main file provided for you with the original main file in your SDK project. Make sure to have a copy of original main file for yourself!

The project is documented at http://processors.wiki.ti.com/index.php/CC32xx_SPI_Demo and in the CC3200 SDK (C:\ti\CC3200SDK_1.4.0\cc3200-sdk\examples\SPI_Demo). You will need to import the project. In this task you are using only master mode. For the SPI master, you should set MASTER_MODE to 1 in main.c:

```
#define MASTER_MODE 1
```

For the SPI slave, you should set MASTER_MODE to 0, but in this lab, you do not need to do this:

```
#define MASTER_MODE 0
```

As described in the documentation, “always execute master application followed by slave application to avoid slave SPI receiving garbage”.

2. Implement OLED interface using SPI

For the SPI interface to the OLED, you will need to use the interface signals shown in the table below. Note that you will use GPIO signals configured as outputs to control the DC (Data/Command), the OLEDCS (OC) and the RESET signals on the OLED. You can choose any unused GPIO signals on the LaunchPad headers P1 or P2. However, check the LaunchPad schematic to verify that the signals are truly available. For example, the GPIO on P2.9 (Dev Pin#55) is the UART0_TX signal and is not available for GPIO when the console window is used. Do not use the LaunchPad's RESET_OUT signal for the OLED RESET since you should be able to reset the OLED independently from the CC3200 LaunchPad. (NOTE: we are using a GPIO signal to control the OLEDCS instead of the dedicated SPI_CS signal because we plan multiple SPI devices on the CC3200's general-purpose SPI port.). In this task, you only need to use the master_main program. OLED will not send back any data. You just need to send a couple of commands to the OLED.

OLED signal	LaunchPad interface
MOSI (SI)	SPI_DOUT (MOSI)
SCK (CL)	SPI_CLK (SCLK)
DC	GPIO
RESET (R)	GPIO
OLEDCS (OC)	GPIO
SDCS (SC)	n.c. (no connection)
MISO (SO)	n.c. (no connection)
CD	n.c. (no connection)
3V	n.c. (no connection)
Vin (+)	3.3V
GND (G)	GND

Adafruit provides a link to an open source graphics library for this display. (See <http://www.adafruit.com/product/1431> for details.) The code is written for Arduino so we need to port it to the CC3200. We will provide modified library functions with the Arduino-specific code removed or commented out. Your task will be to write low-level functions WriteData() and WriteCommand() that use the SPI port to write a data or command byte to the OLED. You should study the SPI example program that you tested in Part 1. The WriteData() and WriteCommand() functions allow you to make use of the higher-level OLED library functions. Look for comments with “TODO” in the Adafruit_OLED.c and test.c files for specific tasks.

Write a test program that mimics the Arduino example test.ino found at <https://github.com/adafruit/Adafruit-SSD1351-library/blob/master/examples/test/test.ino> Your test program for the OLED display should do the following:

- Print the full character-set found in the font table in glcdfont.h.
- Print the string “your full name!” to the display
- Display a pattern of 8 bands of different colors *horizontally* across the full OLED display.
- Display a pattern of 8 bands of different colors *vertically* across the full OLED display.

2

- Call the testlines() function to display diagonal lines.
- Call the testfastlines() function to display a rectangular grid pattern.
- Call the testdrawrects() function to draw a pattern of rectangles.
- Call the testfillrects() function.
- Call the testfillcircles() and testdrawcircles() functions.
- Call the testroundrects() function.
- Call the testtriangles() function.

There should be a small delay between calling each of these functions to allow each pattern to be clearly seen. Your program should be an infinite loop that continually cycles through the specified test patterns.

Verification

Execute all these functions and demonstrate them in your video.

Debugging

To debug your display, you can capture the SPI waveform using the Saleae logic program. It is optional! For this purpose, you should capture the very first packets transmitted from the micro controller. To be more specific, the first few lines of codes in the Adafruit_Init program are the WriteData() and WriteCommand(), and the Data/Command is sent through the SPI to the OLED board. You should confirm these with you code and check how the captured waveform relates to these instructions.

Capture SPI waveforms using a Saleae logic

To use the logic analyzer on your own computer you will need to install the application (version 1.1.15) from <http://www.saleae.com/downloads>. The Saleae User Guide can be found at <http://downloads.saleae.com/Logic+Guide.pdf>.

- When installing the wire harness into the logic analyzer pod, notice that the orientation is shown on the bottom of the pod. **Make sure that the ground symbol on the bottom of the Saleae module is aligned with the grey wire.** Black is data Channel 0, Brown is Channel 1 and so on (the colors follow the color code for resistor values). **You should make the signal connections with both the CC3200 LaunchPad and the Saleae logic probe *unpowered!***



- The Saleae application includes various serial protocol analyzer, including SPI. Turn on the SPI analyzer. To configure the SPI analyzer, click on the gear icon and select Enable Settings from the menu options.
- Capture an SPI waveform using the Saleae logic probe. It should look like this:

3



Deliverables:

1. The video in *.mp4 format showcasing that your program can display your name on the screen, display lines, circles, etc. similar to the expected results video as outlined in the manual. Please explain the basics of the SPI protocol, and how you configured the Launchpad and OLED to communicate using SPI. As stated below, include comments in your code explaining what it does, any changes you made, and why.
2. The video should also capture you (we should see you in the video at some point) while explaining and demonstrating the working of the circuit. If you have made extra connections, please explain them to us through the video. For example, you can ask someone at home to film you or use the selfie camera to record yourself explaining the work and then use the back camera to capture the demonstration of the circuit. You can also create your own PowerPoint slides and walk us through them for explaining the circuit connection and your idea.
3. Try to limit the video to maximum of 5 minutes. The video and audio clarity should be decent enough for us to judge your work.
4. Zip the source files and video and upload it to the canvas, make sure the zip is not locked and the source files can be opened using text editor software. Please comment the code wherever necessary.